



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/594,421

09/26/2006

Nektarios Georgalas

36-2022

3250

23117

7590

06/21/2010

NIXON & VANDERHYE, PC  
901 NORTH GLEBE ROAD, 11TH FLOOR  
ARLINGTON, VA 22203

EXAMINER

JORDAN, KIMBERLY L

ART UNIT

PAPER NUMBER

2194

MAIL DATE

DELIVERY MODE

06/21/2010

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/594,421	<b>Applicant(s)</b> GEORGALAS, NEKTARIOS	
	<b>Examiner</b> Kimberly Jordan	<b>Art Unit</b> 2194	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 23 March 2010.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1 and 3-16 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1 and 3-16 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                       | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>03/23/2010</u> .  | 6) <input type="checkbox"/> Other: _____                          |

### **DETAILED ACTION**

1. This Office action is in response to the amendment filed on March 23, 2010.
2. **Claims 1 and 3-16** are pending and have been examined.
3. **Claims 1 and 3-16** have been amended.
4. **Claim 2** has been cancelled.
5. The objections to the drawings are withdrawn in view of Applicant's amendments to the specification and drawings.
6. The objections to the specification are withdrawn in view of Applicant's amendments.

### ***Response to Amendment***

### ***Information Disclosure Statement***

7. The Information Disclosure Statement filed on 03/23/2010 has been considered. An initialed copy of Form 1449 is enclosed herewith. Unless otherwise noted, items marked through represent items where the text of the reference was not provided.

### ***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2194

9. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

10. **Claims 1-16** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cugola et al.** (*The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS*), hereinafter Cugola, in view of **Papamarkos et al.** (*Event-Condition-Action Rule Languages for the Semantic Web*), hereinafter Papamarkos, and further in view of **Paton et al.** (*Active Database Systems*), hereinafter Paton.

Regarding **Claim 1**, Cugola discloses:

- *ii) component program code executable to provide a program forming part of a distributed software application* (see at least Abstract, “In an eventbased architecture, distributed software components interact by generating and consuming events. An event is the occurrence of some state change in a component of a software system, made visible to the external world. The occurrence of an event in a component is asynchronously notified to any other component that has declared some interest in it. This paradigm (usually called “publish/subscribe,” from the names of the two basic operations that regulate the

Art Unit: 2194

communication) holds the promise of supporting a flexible and effective interaction among highly reconfigurable, distributed software components.”

Distributed applications, along with any application, is formed by processes.);

- iv) *[event reaction rule storage code executable to store, in an updateable store, one or more event reaction rules] which include one or more calls to said one or more procedures in said component program in reaction to the receipt of said event message* (see at least Section 2.1, Paragraph 1, “An AO is an autonomous computational unit performing an application-specific task. Each active object has its own thread of control and interacts with other AOs by explicitly producing and consuming events.”; Page 829, Column 2, Paragraph 1, “The JEDI framework provides programmers with standard classes supporting the implementation of both active and reactive objects (see Section 2.4). The JEDI class used to implement reactive objects (i.e., the ReactiveObject class) exports an abstract method (called processMessage) that is automatically invoked each time the reactive object has to be notified of an event it has subscribed to.”);
- v) *event reaction rule interpretation code executable to operate said computer in accordance with said event reaction rules by making one or more calls to one or more procedures included with said event reaction rules* (see at least Page 834, Column 2, Paragraph 1, “Process entity representatives show a reactive behavior themselves. In particular, they have a state, subscribe to events, and react to them according to rules that define the set of admissible transitions between states.”);

However Cugola does not explicitly disclose, but Papamarkos discloses:

Art Unit: 2194

- *i) component loading code executable to load component program code including one or more callable procedures on said computer, and to store corresponding component procedure interface information for said one or more callable procedures* (see at least Section 2.2, Paragraph 2, “Valid rules are translated into an XML form and are added by the Registration Unit to the Rule Base (which is an XML file). Details about each rule are stored here, including its name, priority, and event, condition and action parts.” The action, or callable procedure, is part of the rule and details about the rules, or the interface information, is stored as seen in Figure 1. The Registration Unit loads the rules in XML form into the Rule Base.);
- *comprising at least two interconnected computers* (see at least Figure 4)
- *iii) event messaging code executable to receive one or more event messages from another of said computers* (see at least Figure 4, communication between machines; Section 1, Paragraph 1, “ECA rules automatically perform actions in response to events provided that stated conditions hold.”);
- *iv) event reaction rule storage code executable to store, in an updateable store, one or more event reaction rules* (see at least Figure 4, Rule Base; Page 14, Paragraph 3, “Whenever a new ECA rule *r* is registered at a peer *P*, it will be sent to *P*'s SP for storage.”; The SP is also updateable according to Page 15, Paragraph 4, “The latter information is gathered and maintained as follows: if a node in the RDF Schema of an SP changes from not having any data in this peer group to having data, or vice versa, this change is notified to all other SPs so that these can

Art Unit: 2194

update the relevant annotation in their RDF Schemas.”) *which include one or more calls to said one or more procedures in said component program [in reaction to the receipt of said event message]* (see at least Section 1, Paragraph 1, “ECA rules automatically perform actions in response to events provided that stated conditions hold.” The action, or procedure, is performed or called in response to meeting the event conditions.);

However Cugola and Papamarkos do not explicitly disclose, but Paton discloses:

- *vi) event reaction rule modification code executable to allow a user to modify said event reaction rules stored in said updateable store while said component program is running and thereby alter the operation of said distributed software application while it is running* (see at least Page 75, Column 1, Paragraph 3, “Although all active DBMSs support creation and deletion of rules, they can differ in the level of Adaptability supported. In some systems it is only possible to change the rules associated with an application by recompiling the application code, and thus the rules can be modified only at compile time. Others support more dynamic run time rule modification, including the ability of rule actions to modify the rule base. Clearly there is a sliding scale of degrees of Adaptability: in the context of the dimensions, any system that allows rules to be created without recompiling application code can be considered to support run time adaptability.”)

Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate Papamarkos’s ECA rule language into Cugola’s event and

Art Unit: 2194

rule system for distributed applications. It would have also been obvious to one of ordinary skill in the art at the time the invention was made to incorporate Paton's dynamically modifiable event rules into Papamarkos and Cugola's event rule system. See Cugola Page 835, Paragraph 1, "A transition is defined by a triple: triggering event, condition, and action. With this respect, transitions are similar to ECA rules in active databases (see Section 5.1 for a brief description of ECA rules)." Cugola shows transitions are similar to ECA rules, which Papamarkos is drawn to, in active databases, which Paton is drawn to. Papamarkos uses ECA rules in active databases which is a common implementation of event based infrastructure. Cugola simply brings that methodology into distributed applications for an improved distributed system where distributed components can effectively interact.

Regarding **Claim 3**, the rejection of Claim 1 is incorporated, and Cugola further discloses:

- *wherein said event reaction rule interpretation code comprises computational reflection code executable to convert method or procedure call data in said event reaction rule into a corresponding method or procedure call for execution (see at least Page 829, Column 2, Paragraph 1, "Upon activation, an AO subscribes to some events and then starts waiting for their occurrence. When one of these events is notified, the AO performs some operations (possibly generating new events and subscribing or unsubscribing to events) and then starts waiting again. Therefore, it executes a standard loop: to wait for any event among those it has subscribed to and then process it. We use the term reactive object to indicate this*



Art Unit: 2194

particular kind of active object.”; Page 832, Section 2.4, Paragraph 1, “JEDI has been implemented as a framework of Java classes.”)

Regarding **Claim 4**, the rejection of Claim 1 is incorporated. However Cugola does not explicitly disclose, but Papamarkos discloses:

- *said event messages are structured in accordance with event schema data accessible to each of said computers* (see at least Section 1, Paragraph 1, “XML and RDF are becoming dominant standards for storing and exchanging information on the World Wide Web. With their increasing use in dynamic applications such as e-commerce and e-learning [9, 10, 14, 15, 1, 19, 16, 22], there is a need for the support of reactive functionality on XML and RDF repositories. Event-condition-action (ECA) rules are a natural candidate for this. ECA rules automatically perform actions in response to events provided that stated conditions hold.” The messages are written in XML in one case. XML is a structured language with schema. As mentioned in the rejection of Claim 1, the transitions are similar to rules in the ECA format, and the event data is accessible to all of the computers in the distributed system.)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 5**, the rejection of Claim 4 is incorporated. However Cugola does not explicitly disclose, but Papamarkos discloses:

Art Unit: 2194

- *said event messages comprise a combination of event data and mark-up data* (see at least Section 1, Paragraph 1, “XML and RDF are becoming dominant standards for storing and exchanging information on the World Wide Web. With their increasing use in dynamic applications such as e-commerce and e-learning [9, 10, 14, 15, 1, 19, 16, 22], there is a need for the support of reactive functionality on XML and RDF repositories. Event-condition-action (ECA) rules are a natural candidate for this. ECA rules automatically perform actions in response to events provided that stated conditions hold.” XML is a markup language that contains both data, in this case event data, as well as markup data.)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 6**, the rejection of Claim 5 is incorporated. However Cugola, Papamarkos, and Paton do not explicitly disclose:

- *said event messages are sent as encoded text* (It would have been obvious to one of ordinary skill in the art at the time the invention was made to allow network data, in this case event messages, to be sent as encoded text. This would increase the security of a networked or distributed system. On an even simpler note, each XML file may specify a specific character encoding, such as UTF-8, used to encode the XML data contained within the file.)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 7**, the rejection of Claim 1 is incorporated, and Cugola further discloses:

- *an event reaction rule modification code is executable to alter the operation of said distributed software application by specifying a method or procedure to be called and the parameters to accompany said method or procedure call* (see at least Page 829, Column 2, Paragraph 1, “The JEDI class used to implement reactive objects (i.e., the ReactiveObject class) exports an abstract method (called processMessage) that is automatically invoked each time the reactive object has to be notified of an event it has subscribed to.”; Page 831, Section 2.2.5, Paragraph 1, “The ability to move running application components across the nodes of a network is currently a hot topic in software engineering research [22]. Mobility can be used to reduce network traffic since applications can be moved (or can autonomously move) close to the resources they need for their execution.”; Section 2.2.5 describes the details of JEDI’s mobility capabilities. Also see Paton as pointed out in the rejection of Claim 1 on a system’s adaptability.)

Regarding **Claim 8**, the rejection of Claim 7 is incorporated, and Cugola further discloses:

- *said specified method or procedure is running on the other of said computers* (see at least Figure 1; Page 827, last paragraph – Page 828, first paragraph, “In particular, the communication among the components of a distributed system may

Art Unit: 2194

involve more than two parties, and may be driven by the contents of the information being exchanged rather than by the identity of information producers and consumers.”)

Regarding **Claim 9**, the rejection of Claim 1 is incorporated. However Cugola does not explicitly disclose, but Papamarkos discloses:

- *said interconnected computers comprise an administration computer having installed thereon graphical user interface code executable to allow an administrator to update said event reaction rules* (see at least Section 2.2, Figure 1. The rule input is input to the user interface. A graphical user interface is one of the more common user interfaces, and it would be obvious to have a main terminal with the user interface for a user to input the rules.)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 10**, the rejection of Claim 1 is incorporated. However Cugola does not explicitly disclose, but Papamarkos discloses:

- *said event reaction rules specify a method or procedure to be carried out in reaction to the reception of an event message* (see at least Section 1, Paragraph 2, “An ECA rule has the general syntax **on event if condition do actions**. The event part specifies when the rule should be triggered, the condition part is a query

Art Unit: 2194

which determines if the database is in particular state, and the action part states the actions to be performed automatically if the condition holds.”)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 11**, the rejection of Claim 10 is incorporated. However Cugola does not explicitly disclose, but Papamarkos discloses:

- *said event reaction rules further specify a condition to be tested, the carrying out of said action being conditional on said condition being met* (see at least Section 1, Paragraph 2, “An ECA rule has the general syntax **on event if condition do actions**. The event part specifies when the rule should be triggered, the condition part is a query which determines if the database is in particular state, and the action part states the actions to be performed automatically if the condition holds.”)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 12**, the rejection of Claim 1 is incorporated. However Cugola does not explicitly disclose, but Papamarkos discloses:

- *each of said computers further stores database management code executable to provide a database store for said rules stored on said computer* (see at least Section 1, Paragraph 2, “ECA rules have been used in many settings, including

Art Unit: 2194

active databases [25, 20], personalisation and publish/subscribe technology [4, 9, 10, 12, 21], and specifying and implementing business processes [3, 11, 15].”)

Therefore one of ordinary skill in the art at the time the invention was made would be motivated to combine the references for the reasons listed above.

Regarding **Claim 13**, the rejection of Claim 1 is incorporated, and Cugola further discloses:

- *each of said computers further stores component program details including names of one or more procedures or methods provided by said component program* (see at least Section 3.1.2, Paragraph 3, “In the viewer shown in Fig. 8, the process is represented in terms of the process entities stored in the State Server. The rightmost window in the figure illustrates the set of process entity representatives of the technology advisor process that will be presented in more detail in Section 3.2, while the leftmost window describes the lifecycle of a particular process entity representative and its current state.”)

Regarding **Claim 14**, the rejection of Claim 13 is incorporated, and Cugola further discloses:

- *said component program details further include names of one or more input parameters to be included with a method call and an indication of the type of those input parameters* (see at least Page 836, Paragraph 1, “In the viewer shown

in Fig. 9, the process is represented in terms of the sequence of activities that constitute the process and of the input-output and controlflow relationships.”)

Regarding **Claim 15**, the rejection of Claim 13 is incorporated, and Cugola further discloses:

- *graphical user interface code executable to enable a user to view said component program details* (see at least Figures 8 and 9; Section 3.1.2, Paragraph 1, “OPSS Viewer is a monitoring tool that provides information on the state of the process...The Viewer collects all these events and exploits them to provide human agents with an initial visualization of the process state. After terminating this initial setup, the Viewer listens to all the events that notify specific state changes occurring during the normal execution of the process, and use their contents to update the information offered to the human agent.”)

Regarding **Claim 16**, the scope of the instant claim does not differ substantially from that of Claim 1. Accordingly, Claim 16 is rejected for the same reasons as set forth in the rejection of Claim 1.

### ***Response to Arguments***

11. Rejection of claims under §103(a):

Applicant’s arguments with respect to Claims 1-16 have been fully considered but are not persuasive.

Art Unit: 2194

Applicant asserts that the prior art of record does not teach that the computer includes code for loading component program code onto the computer, the component program code including one or more callable procedures, nor does it teach storing corresponding component procedure interface information for said one or more callable procedures. However, as shown in the above rejection the art does teach these limitations. As seen in the prior art, the event rules include conditions as well as actions. The actions are interpreted as the callable procedures which are executed or enacted when the conditions are met. As seen in Papamarkos Section 2.2, Paragraph 2, "Details about each rule are stored here, including its name, priority, and event, condition and action parts." These details are interpreted as the interface information which is stored in the Rule Base.

Applicant also asserts that the reflection code limitation in claim 3 is not taught in Cugola since this term relates to an ability of a program to alter itself and a conventional Java class is unable to do so. However, the specification seems to indicate that a Java Reflection API is used for the ability to use events defined while the application is running. This API, per the specification, is provided as part of a Java programming language package provided by Sun. Since Cugola's Action Object is a java class and the Java Reflection API appears to be known in the art it would follow that it would have been obvious for one of ordinary skill in the art at the time of the invention to incorporate the Java Reflection API into Cugola's Action Object.



***Conclusion***

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

13. Any inquiry of a general nature or relating to the status of this application or concerning this communication or earlier communications from the examiner should be directed to Kimberly Jordan whose telephone number is 571-270-5481. The examiner can normally be reached on Monday-Friday 9:30am-5pm EST. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on 571-272-6799.

Art Unit: 2194

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Kimberly Jordan/  
Examiner, Art Unit 2194

/H. S. Sough/  
Supervisory Patent Examiner, Art Unit 2194  
06/20/10